IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF NEW YORK

INTERNATIONAL BUSINESS
MACHINES CORPORATION,

Plaintiff

v.

PLATFORM SOLUTIONS, INC.,

Defendant

Civil Action No. 06 CV 13565 (LAK)

DECLARATION OF DR. YALE N. PATT
IN SUPPORT OF PSI's CLAIM CONSTRUCTION BRIEF

I, YALE N. PATT, declare as follows:

1. I have personal knowledge of the facts set forth herein and, if called as a witness, would testify to the truth of such facts under oath.

2. I am the Ernest Cockrell Jr. Centennial Chair in the Electrical and Computer Engineering Department of the University of Texas at Austin. I teach computer science and computer architecture to both undergraduates and doctoral students, and have been doing so since I was appointed an Assistant Professor at Cornell in 1966. I also have numerous research interests, and have been continuously conducting research, publishing and lecturing in the fields of computer science and computer architecture from the late 1960s through to today. I have attached a copy of my curriculum vitae, which describes my background and qualifications in more detail, as Exhibit 1 to this Declaration.

3. I submit this Declaration in support of Defendant PSI's Claim Construction Brief to address the interpretation and meaning of certain terms used in the claims of United States Patent Nos. 5,987,495; 6,775,789; 5,953,520; 6,009,261, 5,687,106; 5,696,709, 5,825,678; 6,654,812; 6,971,002; and 5,414,851. I also submit this declaration to respond to the Declaration of Dr. Mark Smotherman filed in support of IBM's Opening Claim Construction Brief, and to provide expert testimony regarding the level of knowledge of a person of ordinary skill in the art at the time of the inventions of the patents-in-suit.

4. In connection with the preparation of this declaration, I have reviewed the patents-in-suit, the prosecution file histories for the patents-in-suit, prior art cited during prosecution, IBM's Opening Claim Construction Brief and supporting declarations, and other references cited in this Declaration. I have also consulted with other experts and colleagues to hear their ideas and memories of what various terms meant to them at the relevant time.

I. Processor

5. I agree with Dr. Smotherman that a computer system cannot function without a processor that interprets and executes instructions. Dr. Smotherman's description of the way a processor interprets and executes these instructions, however, is incomplete. In general, the processor performs the following steps when interpreting and executing an instruction. First, the

processor *fetches* the instruction, meaning that the instruction is loaded into the processor. During this step, the processor also updates the *program counter*, which is a number held in a register in the processor that identifies the next instruction the processor must execute. Next, the processor decodes the instruction by identifying the *opcode* (which, in turn, identifies the operation to be performed) and the *operand specifiers* (which identify the data upon which to perform this operation), if any. Next, the processor evaluates any addresses that are required to execute the decoded instruction—meaning that it figures out what location the values in the operand specifiers are pointing to as containing the data the processor should act upon. The processor next fetches that data and then executes the instruction by applying the opcode to the data. Finally, the processor stores the result of the operation in the location indicated by one of the operand specifiers. The processor then begins this process for the next instruction, which is indicated by the updated program counter.

6. To a person of skill in the art at the time of the invention of the patents-in-suit (*i.e.*, the 1990s), the term "processor" referred to a set of components that were built into one or more integrated circuits. In the case of most computers, the "processor" was built into a single integrated circuit (a "chip" or "microchip"), while in other cases the "processor" included one or more circuit boards, each containing more than one integrated circuit. In both cases, the "processor" consisted of *hardware*. At no time has the term "processor" had a meaning which encompassed software.

7. I agree with Dr. Smotherman that microcode performs very simple operations as part of the interpretation and execution of instructions. I disagree, however, with the remainder of Dr. Smotherman's contentions. Microcode is not "software"—it is part of the hardware of the computer. Microcode has some similarities to software (namely the fact that it can be written out in code), but it is a *manufactured component* of the hardware, not a software program that can be loaded and/or modified (which is what makes software "soft"). To a person of skill in the art at the time of the invention in the 1990s and all the way through today, microcode has been stored in unchangeable circuits *within* the larger integrated circuits that make up the processor. Indeed, if you receive (for example in the mail from Intel) a microcoded x86 processor and you were to

somehow *remove* the microcode (for example, by cutting out the read-only circuits that store it), you would no longer have an x86 processor. The point is this: microcode is seen by people in the art as being a manufactured component of the hardware because, unlike software, it cannot be changed and instead comes burned into the chip like any other logic element.

8. Sometimes, as Dr. Smotherman notes, microcode is stored on ROM or RAM. But ROM and RAM, as Dr. Smotherman would certainly agree, are *integrated circuits*, not software. Dr. Smotherman's reference to "ROM or RAM separate and apart from the integrated circuit(s)" is, therefore, nonsensical.

9. I agree with Dr. Smotherman that, in the 1960s, microcode was stored outside integrated circuits. Each of the references Dr. Smotherman identifies in paragraph seven of his report, however, dates to the late 1960s, with the *most recent* system dating to 1970. But the technology of this time period is completely irrelevant to how a person of ordinary skill in the art would understand the technology referenced in patents written in the 1990s and later. Indeed, in the 1960s, processors mostly consisted of hardware *other* than integrated circuits and the first microprocessor, the Intel 4004, was not invented until 1971. But computer technology, and the common usage of the terms that describe it, has changed fundamentally since that time. In fact, integrated circuit technology has developed so dramatically that rather than 2,300 transistors on a chip, as was the case in 1971, manufacturers were putting *3,100,000* transistors on a chip as early as 1992 (the Intel Pentium).

10. The forms of storage Dr. Smotherman identifies—magnetic core memory, mylar cards, and discrete capacitors—are archaic, and have not been in widespread use since the early 1970s. They were certainly archaic at the time the patents-in-suit were invented, in the 1990s and thereafter. Indeed, those forms of storage were not even in use at the time the patents in suit were invented. And, if you *had* tried to store microcode in them in a 1990s model computer, you would have produced a product so absurdly slow that no one would have considered marketing it.

11. Dr. Smotherman's discussion of "millicode" is an irrelevant distraction. None of the patents-in-suit refer to "millicode," and "millicode" does not make up part of a processor.

"Millicode," as its name implies, operates at a level ***outside*** the processor. It implements highly efficient routines for frequently called functions."[1] An article in the IBM Journal of Research and Development called "Millicode in an IBM zSeries Processor"[2] makes this clear. Far from making up part of the processor, it is *software* "written in a low level language," made up of instructions that must be interpreted and executed just like any others. As IBM puts it, "in many ways, millicode is handled by the processor hardware similarly to the way operating system code is handled....Millicode execution uses the same basic data flow as is used to execute system instructions."[3] Millicode is software that *runs on* the processor; it is not *part of* the processor.

II. Instructions and Instruction Sets

12. I agree in large part with Dr. Smotherman's statements in paragraph nine of his declaration. An instruction set architecture, or ISA, defines the specific set of instructions that a particular processor can directly execute. Thus, an instruction set is simultaneously the set of instructions in an ISA, and the set of instructions that can be executed by a processor built to implement that architecture. Processors cannot execute instructions written for an ISA other than the one to which the processor was designed; they view instructions from an alien ISA as gibberish. An "instruction" is a string of digits that can be directly executed by the processor to which it is directed. As to all other processors, that string of digits is treated as data to be acted upon, not as an instruction to be executed. If one processor wants to emulate the instruction set of another processor, it must first use software to translate this data into instructions that the emulating processor can understand. To put it another way, instructions are not instructions in the abstract, but are instead instructions *relative to* a particular processor.

13. I disagree with Dr. Smotherman's assertion that there are different "levels" of instructions. A computer architecture includes only one kind of instruction, sometimes redundantly referred to as "machine instructions" or "computer instructions." These are strings of digits that can be directly executed by the processor to which they are directed.

[1] *See, e.g.*, *What the Heck Is Millicode?*, Byte (Dec. 1995), *at* http://www.byte.com/art/9512/sec14/art3.htm.

[2] *Available at* http://www.research.ibm.com/journal/rd/483/heller.html.

[3] *Id.*

14. Dr. Smotherman refers to "assembly language instructions" as another kind of instruction. It is true that assembly language statements are similar in many ways to machine instructions. But there also important distinctions. First, assembly language statements are usually made up of four parts: a label (optional), an opcode expressed symbolically in English, operands expressed symbolically in English, and a comment (optional). Before it can be processed by a computer, it must first be translated into a series of digits. The comment is removed, since it has no relevance to the processor. The label is used by the translator in the translation process and then discarded. At the end of this translation process, the thing that is left, the thing that is used by the computer, is an instruction (a.k.a. a machine instruction). In contrast, an assembly language "instruction" cannot be fetched from memory, nor can it be decoded or executed. So, in the context of a processor fetching, decoding, and executing instructions, it is incorrect to say that there are two alternatives, machine instructions and assembly language instructions—only "instructions" or "machine instructions" apply.

15. I agree with Dr. Smotherman that the assembly language statement "ADD X, Y" to add two numbers together corresponds to, and would be translated into, the string of digits making up an instruction (*e.g.*, "0001001001000010"). I do not agree, however, that these are each "language constructs." The term "language construct" is not a term used in the art to refer to instructions, but instead is used to refer to syntactically allowable statements in programming languages such as C and Java. In any event, a person of ordinary skill in the art would understand that the patents at issue are discussing instructions *as they exist within the computer* – namely in their object code form as a string of ones and zeros – because this is the only form in which they can do any work.

16. Finally, I disagree with Dr. Smotherman's conclusion that one of ordinary skill in the art at the time the patents were filed would not have understood "instruction" to mean something that could be directly executed by a processor, and that only "machine instruction" would have had that meaning. The terms "instruction" and "machine instruction" are synonymous in the field of computer architecture. One might refer to an instruction as a "machine instruction" when one needed to distinguish it from concepts outside the field of

computer architecture that go by the name "instruction," such as statements in some higher-level programming language. But, in the context of the patents, such higher-level programming languages are irrelevant – because the patents all are aimed at the actual functions of a computer system. In that functional context, the only form in which something is recognized *by the computer* as being an instruction is when it is a series of electrical voltages that represent 1s and 0s. For this reason, to one of ordinary skill in the art at the time the patents-in-suit were invented, the term "instruction" as used in the context of the asserted patents refers precisely and unambiguously to a string of digits that specifies an operation and identifies its operands, if any, and can be directly executed by the processor to which it is directed.

III. "Program Status Word"

17. A program status word ("PSW") is the contents of the register that directs the processor in the execution of a program. I agree that this data may include the address of the next instruction to be executed (as is the case with the IBM zArchitecture), the condition codes, and the program authority—although the PSW in a particular architecture might not include each of these, or might include additional data.

18. I disagree with Dr. Smotherman when he says that the PSW could be stored in a location other than in a dedicated register. A person of skill in the art in the 1990s and thereafter would not regard the PSW as something that could be stored in, as Dr. Smotherman contends, a "magnetic core." Indeed, the System/360 Model 40 to which Dr. Smotherman refers was introduced in *1964* and has not been sold by IBM since *1977*—two decades before the patents-in-suit were invented.[4] As noted above, magnetic core memory would not even have been *functional* for use to store the PSW in the 1990s and thereafter. Indeed, it was considered archaic even in the 1970s and 80s. Anyone with even minimal skill in the art of computer architecture at the time the patents-in-suit were invented would have known that a computer whose PSW was not stored in a hardware register in the processor would not be able to execute programs at a reasonable speed to be a viable product. The PSW frequently has to be examined multiple times during the execution of each instruction, and registers are the *only* computer

storage fast enough to allow these examinations to occur during the execution of a single instruction. With all due respect, Dr. Smotherman's assertion to the contrary is silly.

19. The PSW is stored in a dedicated register, and the term "program status word" would be understood by one of ordinary skill in the art to refer *to the contents of that register*. It is, after all, the contents of that register that *defines* what the computer is to do next and the operating authority of the current program. If a particular piece of data is not in the register to which the processor looks when it wants to know what instruction to execute next, it *can't* be the program status word because it *can't* perform the function of the program status word. Whatever data is in that register *is* the data that defines what instruction the processor will perform next, the operating authority of the current program, and so on. The contents of that dedicated register, therefore, is by definition the PSW.

20. While it is certainly possible to *save* the information from the PSW to a location in memory, when this is done, the saved information is just data. The computer does not see it as a program status word, and does not treat it as such. It is, instead, a record of what the program status word *was* at a prior point in time—not the actual PSW.

IV. Registers

21. As used in the patents, a register is a hardware storage element in the processor that can be accessed substantially faster than memory. It is not, as Dr. Smotherman contends, *any* location of an operand in a restricted set of storage locations. Once again, Dr. Smotherman refers to the IBM System/360 from 1964, which used magnetic core memory outside the processor as its registers. But as noted above, whether registers could practically have been stored outside the processor in a *1964* mainframe is irrelevant to the meaning of the term "register" as of the 1990s and beyond, which is at issue here. By the 1990s (and, indeed, by the late 1970s), magnetic core memory was much too slow, compared to the then-prevailing processor speeds, to use for registers. At the time the patents were invented, it would have been *ludicrous* to suggest placing registers outside the processor.

22. For this reason, by the time the 1990s came around, the term "register" had taken

[4] *See* http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframePP2040.html

on a different meaning from the one it had back in the 1960s. From a point well before the 1990s, the term register referred specifically to a hardware storage element *in* the processor that (among other qualities) could be accessed much faster than memory. That is what anyone reading the patents in suit at the time of their invention would have understood the term to mean – a meaning that is equally plain from both the language and the drawings of the patents in suit.

V. Floating Point Unit

23. I agree with Dr. Smotherman that computers often perform calculations (which Dr. Smotherman more generally refers to as "operations") with floating point numbers. That portion of a processor's circuitry that performs floating point calculations is called the "floating point unit." The patents refer to some of this circuitry as "FPU control logic," "logic" being a synonym in this context for circuitry. Some computers, however, lack a "floating point unit," and must instead use software to perform floating point calculations. Such computers are rare these days—even the tiny ARM processors used in many mobile phones now frequently contain a floating point unit.[5] Thus, I agree with Dr. Smotherman that floating point ***calculations*** can be performed either in hardware or in software. But I note that a floating point ***unit*** is a portion of the processor's circuitry that performs floating point calculations, and the term "floating point unit" does *not* encompass software.

24. I agree with Dr. Smotherman that many floating point operations occur with numbers in "normalized" form. I disagree, however, with his conclusion that floating point operations on numbers in normalized form were commonly performed in hardware, whereas floating point operations on numbers in denormalized form were commonly performed in software. Whether none, some, or all floating point operations could be performed in hardware rather than software was, at the time of the filing of the '106 patent, entirely a function of the price point of the relevant processor. As noted above, some processors had no floating point unit; other processors had a floating point unit that could deal only with normalized numbers, leaving operations on denormalized numbers to be performed outside the floating point unit, by software. Some processors could perform operations on denormalized numbers in the hardware

of their floating point units. Indeed, such a processor was described in United States Patent No. 5,058,048, titled "Normalizing pipelined floating point processing unit," issued October 15, 1991, four years before the filing of the '106 patent.

25. I further note that Dr. Smotherman incorrectly asserts that software carrying out floating point instructions constitutes part of the "floating point unit." This is not the case: a floating point unit is *circuitry*—a part of the processor—and not software. Floating point operations can be carried out in software, but that does not make the software a "floating point unit."

VI. Host-Network Interface

26. I agree with Dr. Smotherman that computers can be partitioned into sections that act as independent computers. I also agree that, as of 1998, partitions often used a host-network interface to connect to a network and that, in 1998, several IBM systems used a host-network interface consisting of an OSA adapter card connected to the computer via an ESCON interface. The ESCON interface, however, is a specific way (called an optical serial connection) of connecting peripherals to an IBM mainframe system—it is not software. Similarly, the OSA adapter card is, as its name implies, a **card**—a piece of hardware, which (like most hardware) has software that runs on it.

27. But none of this means that the host network interface is simply any "hardware and software" that "interfaces" between the computer and the network. Instead, people of ordinary skill in the art would understand that the interface is a hardware *component* that makes use of software stored on a memory *inside the component* to perform its task. It cannot be properly construed as constituting any hardware or software involved in sending signals to and from the network because, as a practical matter, this would include everything in a computer system. Instead, the host-network interface is a discrete hardware component that is physically connected to the network port and (through a combination of physical circuits and the use of software stored in a memory within it) performs the tasks needed to send and receive signals coming and going directly from and to the network port.

[5] *See* http://www.embedded.com/story/OEG20010612S0080

28. I further note that the OSA adapter card contains memory on which the list of IP addresses belonging to the various partitions are stored. That the IP addresses are said in the patent to be stored "at the host-network interface" means that they are stored in this memory.

VII. Firmware Image

29. The term "firmware" has no single fixed meaning to persons of ordinary skill in the art. Indeed, I cannot think of a term that has been used (or misused) more and applied to more different things in the field of computer architecture. Firmware started out meaning microcode (in the 1960s). But by the time the patents in suit were filed in the 1990s, people regularly used the term to refer to code stored in all kinds of different media, with different degrees of protection from erasure, and with different relationships to the various other architectural layers in a computer system. As a result, the word "firmware" – in isolation – had no specific meaning to a person of ordinary skill in the art at the time of the invention of the patents in suit.

30. For this reason, a person of ordinary skill in the art at the time of the invention would have had to deduce the meaning of the term "firmware" from whatever meaning it was given in the patent. In performing this deduction, a person of ordinary skill in the art would have given great weight to the explicit statement in the patent that "**Firmware is** 'software' stored in a memory ***chip*** that holds its content without electrical power..."[6] '002 patent at 6:22-24 (emphasis added). Although this is not by any means the only possible meaning one could give to the term, it is apparently the meaning given to the term by the author of the patent.

31. I agree with Dr. Smotherman that for most computers firmware is used during the computer's start-up process. But this is certainly not the only time when firmware is used – because firmware, like software, can perform a wide variety of functions. For this reason, persons of ordinary skill in the art do not define firmware in terms of what it *does*—because what it does is variable from system to system and component to component

32. I agree with Dr. Smotherman that firmware can be copied to some other storage location that is not "a memory chip that holds its content without electrical power," and that such

a copy could be a "firmware image." However, I disagree with Dr. Smotherman's contention that a person of ordinary skill in the art, upon reading the '002 patent, would think that the firmware that was being copied could be stored anywhere other than "*in a memory chip* that holds its content without electrical power." This is because, as I stated in the previous paragraph, the term "firmware" has no fixed meaning and because a person of ordinary skill in the art would therefore have to use whatever meaning was given in the specification. Although I would not have defined the term that way, the specification clearly states that "firmware" can only be stored "in a memory chip that holds its content without electrical power." The different storage types listed by Dr. Smotherman, such as floppy disks and CDs, could not store "firmware" as that term is used in the '002 patent, because they are not "a memory chip that holds its content without electrical power."

33. Similarly, while the programs on the floppy disks that helped the IBM System/370 Model 145 (to which Dr. Smotherman refers) boot **in *1970*** might be able to be referred to as "firmware" in some broad sense of the term as it was then used, that is not the sense in which it is used in the '002 patent (filed ***31 years* later**). As I mentioned earlier, by the time the patents in suit were filed, the term had lost whatever meaning it had in the 1960s and 70s—and even in the 1960s and 70s "firmware" had no specific meaning. Accordingly, a person of ordinary skill in the art would simply have looked to the definition given in the specification – a definition that is certainly an acceptable use of the term.

VIII. Routine

34. I agree with Dr. Smotherman that the term "routine" is used in some contexts to refer to sections of source code that a programmer can use repeatedly instead of having to rewrite the same code again and again. But in the context of the emulation patents, the code at issue is clearly *not* source code: instead, it is machine code, made up of "target" (or "native") instructions.

35. Moreover, these "routines" must be executed in sequence and retired in order. If a programmer codes a routine to display a web page, for example, the routine must **first**

[6] '002 patent at 6:22-24 (emphasis added).

download the HTML and image data from the web server, and **then** display that information on the screen. If, as Dr. Smotherman seems to argue, a routine has no particular order, then the program might attempt to display the information before it is downloaded from the server, which obviously couldn't work.

36. Dr. Smotherman is correct, of course, that adding 1 to 2, then adding 3 gives the same result as adding 3 to 2, then adding 1. This is because, as a middle school math teacher would say, addition displays the associative property. But, as the example given in column 4 of the '520 patent makes clear, even the process of adding numbers requires a *sequence* of instructions. In particular, the processor *must* load the data representing the numbers *before* it adds those numbers together, and it must add the numbers together *before* it stores the result. Thus, even in Dr. Smotherman's simple example, a semantic routine to emulate an instruction to add 1+2+3 would have to have an internal order and could not fairly be characterized as simply being a "set" of instructions. In short, it is universally understood in the art that computer instructions have *dependencies*, and that a routine must have an internal order. This concept is so elementary that it would strike one of ordinary skill in the art as ridiculous to suggest that the instructions in a routine should not be understood to have an internal order.

37. Furthermore, as noted at the outset of this declaration, a processor actively chooses the next instruction to execute based upon the contents of the updated program counter. That program counter is updated in each instruction cycle based on the results of the last cycle and the information in the decoded instruction. Thus, the *mechanism* by which the order of instructions are executed in a computer actually *precludes* a random order. If you feed the instructions into the processor in a random order, you either end up crashing the computer or creating gibberish. Dr. Smotherman refers to "conditional logic" to argue that instructions within routines do not always execute in the same order. This is true, but it is also *irrelevant*. Take a source code routine, for example, that includes a sequence of instructions like "(1) add the value in register A to the value in register B and, (2) if the result is positive, (3) sound the alarms." Of course, the instruction "sound the alarms" may or may not be executed, depending on the values in register A and register B. But that does not mean that the instructions are a mere "set" that has

no order. If that were true, there would be an awful lot of false alarms when the system chose to execute the third instruction before executing the first two.

38. The point is simply that the concept of a sequence, to persons of ordinary skill in the art, does not require a *single* fixed order. Instead, the concept encompasses complex internal orders *including* branches and conditional events. Thus, there is no conflict between the presence of conditional logic and the notion that the phrases "semantic routine" and "target routine" imply that their elements have an internal *sequence*. Indeed, going in the other direction, as Smotherman suggests, and reading *all* internal order out of the concept of a "routine," results in both technical and logical absurdity.

IX. Additional Items

39. All computer systems store information in memory. Memory is organized into a specific structure, and each memory location has its own "memory address." Calculating a memory address is a different operation from accessing the information stored at that memory address. This is similar to finding one's way to a store: looking up the store's address is different from going to the store, and one must first know the address before one can go to the store.

40. In general, computer systems run applications *after* the operating system is up and running. This is because the operating system provides a mechanism to manage the computer's resources and act as a referee among various application programs, which will compete for those resources. While one can create an application that will stand alone without an operating system running underneath it, doing so is uncommon. In the context of the '002 patent, a person of ordinary skill in the art would understand that the "power on process to boot" the computer system described therein ends with the execution of the partition firmware, as shown in figure 4.

41. In addition, a person of ordinary skill in the art would understand that the "power on process to boot" would end before applications are ready to begin executing because the claims describe the power on process as taking place *before* the operating system is booted and because (except for applications that run on a bare machine, which would be very unusual on a mainframe) the operating system must be booted before applications are ready to begin executing.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct, and that this Declaration was executed on June 4, 2008 at Barcelona, Spain.

YALE N. PATT

CURRICULUM VITAE - Yale N. Patt

PERSONAL

Yale N. Patt
901 W. 9th Street, Suite 603, Austin, TX 78703
(or, Electrical and Computer Engineering Department, The University of Texas at Austin,
1 University Station C0803 Austin, TX 78712-0240)

EDUCATION

Northeastern University, B.S. in Electrical Engineering, June 1962

Stanford University, M.S. in Electrical Engineering, June 1963

Stanford University, Ph.D. in Electrical Engineering, June 1966

PRIMARY EMPLOYMENT (Since obtaining the PhD)

Professor of Electrical and Computer Engineering, and Ernest Cockrell, Jr. Centennial Chair in Eng'g, U of Texas at Austin	7/99-present
Professor of Electrical Eng'g & Computer Science, University of Michigan	11/88-6/99
Visiting Professor, Elec. Eng'g & Computer Science, U.C. Berkeley	9/79-8/88
Professor of Computer Science and Mathematics, S.F. State University	8/76-6/89
Associate Professor of Computer Science and E.E., N.C. State University	1/69-5/76
Captain, U.S. Army, assigned to U.S. Army Research Office	6/67-6/69
Assistant Professor of Electrical Engineering, Cornell University	9/66-6/67

HONORS AND AWARDS

IEEE/ACM Eckert-Mauchly Award, 1996, "for important contributions to instruction level parallelism and superscalar processor design."

2000 ACM Karl V. Karlstrom Outstanding Educator Award, "for great ability, dedication, and success in developing computer science education, and for outstanding achievements as a teacher."

1995 IEEE Emanuel R. Piore Award (the IEEE Technical Field Medal for Information Processing), "for contributions to computer architecture leading to commmercially viable high performance microprocessors."

1999 IEEE Wallace W. McDowell Award, "for your impact on the high performance microprocessor industry via a combination of important contributions to both engineering and education."

2005 IEEE Charles Babbage Award, "for fundamental contributions to high performance processor design."

Fellow, IEEE, for "Innovative contributions to the implementation of high performance computer architectures."

Fellow, ACM, for "many outstanding seminal contributions to high performance microarchitecture and for leadership and teaching in computer science and engineering education.

IEEE/ACM 2007 ISCA Most Influential Paper of the Year Award, for "the paper published in ISCA in 1992 that has had the most impact on the field (in terms of research, development, products or ideas) during the 15 years since it was published." For: "Alternative Implementations of Two-Level Adaptive Branch Prediction."

IEEE Third Millenium Medal, April, 2000.

IEEE Computer Society Golden Core Award, 1996.

Outstanding Lecturer of the Year, National ACM Lectureship Program, 2000-01.

Outstanding Lecturer of the Year, National ACM Lectureship Program, 1998-99.

Texas Excellence Teaching Award, presented by the Texas Exes (the Univerity's Alumni Association), 2002.

Dad's Association Centennial Teaching Fellowship, The University of Texas at Austin, Fall semester, 2002.

Appointed Arthur F. Thurnau Professor at the University of Michigan, March, 1998, "for his excellence in teaching and his dedication to undergraduate students."

Member, Sigma Xi, Tau Beta Pi, Eta Kappa Nu Honor Societies.

National Tau Beta Pi Fellow, 1962-1963.

1997 Research Excellence Award, University of Michigan, Dept of Electrical Engineering and Computer Science.

1996 Outstanding Teacher Award, University of Michigan, College of Engineering.

1995 Teaching Excellence Award, University of Michigan, Dept of Electrical Engineering and Computer Science.

Outstanding Professor of the Year, 1991-1992, selected by the Michigan student chapter of Eta Kappa Nu.

Adaptive Insertion Policies for High-Performance Caching. *IEEE Micro,* Top Picks Special Issue, January/February, 2008. (with Moinuddin K. Qureshi, Aamer Jaleel, Simon C. Steely Jr., and Joel Emer).

Diverge-Merge Processor: Generalized and Energy-Efficient Dynamic Predication, *IEEE Micro,* Top Picks Special Issue, January/February, 2007, (with Hyesoon Kim, Jose Joao, and Onur Mutlu).

Techniques for Efficient Processing in Runahead Execution Engines. *IEEE Micro,* Top Picks Special Issue, January/February, 2006, (with Onur Mutlu and Hyesoon Kim).

Wish Branches: Combining Conditional Branching and Predication for Adaptive Predicated Execution, *IEEE Micro,* Top Picks Special Issue, January/February, 2006, (with Hyesoon Kim, Onur Mutlu, and Jared Stark).

Runahead Execution: An Effective Alternative to Large Instruction Windows, *IEEE Micro,* Top Picks Special Issue, November/December, 2003. (with Onur Mutlu, Jared Stark, and Chris Wilkerson).

Best paper award in Hardware-Architecture at the 22nd Annual HICSS conference, January, 1989, "Tailoring Functional Units and Memory in a High Performance Prolog Architecture."

Best paper award in Hardware-Architecture at the 21st Annual HICSS conference, January, 1988, "HPSm2: A Refined Single Chip Microengine."

Best paper award in Hardware-Architecture at the 20th Annual HICSS conference, January, 1987, "A Clarification of the Dynamic/Static Interface."

Best paper award in Hardware-Architecture at the 19th Annual HICSS conference, January, 1986, "An HPS Implementation of VAX: Initial Design and Analysis."

KEYNOTE ADDRESSES

Keynote address, European Union HiPEAC Summer School for PhD students in Computer Architecture and Embedded Systems (ACACES), July 2008, L'Aquila, Italy. Title: The Multi-core Era: What does it mean? (and even more importantly, what does it NOT mean?).

Keynote address, International Symposium on Systems, Architectures, MOdeling and Simulation (SAMOS VIII), July 2008, Samos.

Keynote address, HiPC Symposium on High Performance Computer Architecture, Goa, India, December 2007. Title: The Transformation Hierarchy in the Era of Multi-Core.

Keynote address, 25th International Conference on Computer Aided Design (ICCD), Squaw Valley, October 2007. Title: Microprocessor Performance, Phase II: Harnessing the Transformation Hierarchy.

Keynote address, Second Spanish Conference on Informatics (CEDI 2007), Zaragoza, September, 2007. Title: The Microprocessor Ten Years from Now: Why it is Relevant to all Informatics.

Keynote address, 16th International Conference on Parallel Architecture and Compiling Techiques (PACT), Brasov, Romania, September 2007. Title: Harnessing the Transformation Hierarchy.

Keynote address, Workshop on Computer Architecture Education (WCAE 2007), in conjunction with ISCA, San Diego, June, 2007. Title: LC-3, x86, or ??: The First ISA for Students to Study.

Keynote address, Student Congress, Durango, Mexico, September 2006.

Keynote address, 33rd Annual IEEE/ACM International Symposium on Computer Architecture (ISCA), Boston, June, 2006. Title: Computer Architecture Research and Future Microprocessors: Where do we go from here?

Keynote address, 2006 Spring convention of the State of New Jersey High School Science and Math Teachers, Newark, March, 2006. Title: Education: Some thoughts after 35 years in the trenches.

Keynote address, 4th Annual Dalhousie Computer Science In-House Conference, Halifax, Nova Scotia, September, 2005. Title: The Microprocessor: Its Characteristics Ten Years from Now.

Keynote address, 4th annual Encuentro Estudiantil (National ACM Congress of Mexican Computer Science Students), Puebla, March, 2005. Title: Are there any questions?

Keynote address, 3rd IEEE/ACS International Conference on Computer and System Applications (AICCSA-05), Cairo, Egypt, January, 2005. Title: The Microprocessor of the Year 2014: Do Pentium 4, Pentium M, and Power 5 provide any hints?

Keynote address, 16th Symposium on Computer Architecture and High Performance Computing, Foz du Iguazu, Brazil, October, 2004. Title: The Microprocessor of the Year 2014: Do Pentium 4, Pentium M, and Power 5 provide any hints?

Plenary address, XXIX Pan-American Engineering Symposium, Mexico City, September, 2004. Title: Future Trends in Computer Architecture.

Plenary address, 2004 International Supercomputer Conference, Heidelberg, June, 2004. Title: "The processor in 2014: What are the challenges, how do we meet them?"

Keynote address, 2004 IEEE International Symposium on Performance Analysis of Systems and Software, Austin, March, 2004. Title: Performance Analysis: A big plus, or an even bigger minus.

Keynote address, 10th annual IEEE International Symposium on High Performance Computer Architecture (HPCA-10), Madrid, Spain, February, 2004. Title: Microarchitecture: Are we finally done?

Keynote address, HiPC 2003, Hyderabad, India, December, 2003. Title: The High Performance Microprocessor in the Year 2013: What will it look like? What it won't look like?

Keynote address, International Conference on Information Technology (CIT), Bhubaneswar, India, December, 2003. Title: Current bottlenecks to continued high performance microprocessors, and what we can do to get passed them.

Keynote address, SSGRR 2003s, L'Aquila, Italy, July, 2003. Title: A Microarchitect's perspective: What computer technology will provide and won't provide by the year 2020.

Keynote address, Computer Architecture Education Workshop, International Symposium on Computer Architecture (ISCA), June, 2003. Title: Teaching and Teaching Computer Architecture: Two Very different topics (Some Opinions about each).

Keynote address, 2nd annual Encuentro Estudiantil (National ACM Congress of Mexican Computer Science Students), Guanajuato, May, 2003. Title: What is Moore's Law REALLY all about?

Keynote address, ICCD, Austin, TX, September, 2001. Title: There is still plenty of work that needs to get done.

Keynote address, 26th EuroMicro Congress, Maastricht, Netherlands, September, 2000. Title: Higher and Higher Performance Microprocessors: Are the Problems Just too Hard to Solve?

Plenary address, EuroPar-2000, Munich, Germany, August, 2000. Title: Despite the Nay-Sayers to the Contrary, Moore's Law is Alive and Well and Still Providing Opportunities.

Keynote address, Seventh International Conference on Advanced Computing, Roorkee, India, December, 1999.

Keynote address, International Symposium on New Trends in Computer Architecture, Gent, Belgium, December, 1999.

Keynote address, 11th Symposium on Computer Architecture and High Performance Computing, Natal, Brazil, September, 1999.

Keynote address, Workshop on Workload Characterization, 31st Annual Symposium on Microarchitecture, Dallas, TX, November 1998.

Keynote address, Workshop on Simultaneous Multithreading, 4th Annual High Performance Computer Architecture Conference (HPCA-4), Las Vegas, January, 1998.

Keynote address, Computer Architecture Education Workshop, 4th Annual High Performance Computer Architecture Conference (HPCA-4), Las Vegas, January, 1998. Title: High Tech -- The Enemy of Education?

Keynote address, IEEE/ACM 11th Annual International Supercomputer Conference, Vienna, July, 1997.

Keynote address, IEEE International Performance Conference on Computers and Communications, Phoenix, Arizona, February, 1997.

Keynote address, IBM Conference on Performance Analysis and its Influence on Computer Design, Austin, March 27, 1996.

Plenary address, 29th Annual Hawaii International Conference on System Sciences, Maui, January, 1996. Title: The Microprocessor and its Performance in the Year 2000.

Plenary address, ACM/IEEE International Conference on Parallel Architectures and Compilation Techniques (PACT95), Limassol, Cyprus, June, 1995.

Keynote address, International Congress on Massively Parallel Information Systems, Ischia, Italy, May, 1994.

Keynote address, International Federation of Information Processing (IFIP) Working Conference on Architectures and Compilation Techniques for Fine and Medium Grain Parallelism, Orlando, Florida, January, 1993. Title: The Compiler and the Microarchitecture: Partners, not Adversaries.

Keynote address, 25th Annual IEEE/ACM International Symposium and Workshop on Microarchitecture, Portland, OR, November, 1992.

Keynote address, Euromicro Congress 1992, Paris, September, 1992.

Plenary address, 25th Annual Hawaii International Conference on System Sciences, Kauai, January, 1992. Title: What do we do about the Speed of Light?

Keynote address, 5th International Congress on Computer Architecture and Performance Modeling, Torino, Italy, February, 1991. Title: "Computer Architecture and Performance Modeling, Obtaining Meaningful Information."

Keynote address, NCR Corporation Workshop in Performance Measurement, Cambridge, OH, December, 1988. Title: "The light is better over here."

Keynote address on "Computer Architectures for Artificial Intelligence," IEEE 1987 Symposium on Artificial Intelligence: Applications in Engineering, Minneapolis, Minnesota, January, 1987.

Keynote address (Architecture) at the annual Livermore-Los Alamos invited Salishan Workshop on "Toward a Science of Parallel Computation," March, 1986.

DISTINGUISHED LECTURES

Lecturer, Pennsylvania State University Distinguished Lectures Series, April, 2007. Title: The Fundamentals: Vehicle to the Year 2017- Tomorrow's Graduates, Tomorrow's Microprocessors

Lecturer, The University of Arizona Distinguished Lectures Series, February, 2007. Title: The Microprocessor Ten Years from Now: What are the Challenges? What can we do about them?

Lecturer, The European Union's 2nd Annual HiPEAC Summer School for PhD students, July, 2006, L'Aquila, Italy. Five 90 minute lectures on the subject, Advanced Microarchitecture.

The Saul Gorn Memorial Lecturer, University of Pennsylvania, April, 2006. Title: Computer Architecture Research: Is it Dead? Is it in need of Revitalization? Where do we go from here?

Lecturer, The University of Nebraska Distinguished Lecturer Series, April, 2006. Title: The Microprocessor Ten Years from Now.

Lecturer, The Northeastern University Distinguished Lecturer Series, April, 2006. Title: Future Microprocessors, where do we go from here?

The Irons Distinguished Lecturer, Rutgers University, March, 2006. Title: Future Microprocessors, where do we go from here?

Lecturer, The European Union's Inaugural HiPEAC Summer School for PhD students, July, 2005, L'Aquila, Italy. Five 90 minute lectures (Monday through Friday) on the subject Advanced Microarchitecture.

Lecturer, Distinguished Lecturer Series, SMU, April, 2005. Title: "The Microprocessor in the Year 2015: Issues, Challenges, Potential Avenues to Solutions."

Lecturer, ISTeC Distinguished Lecturer Series, Colorado State University, Ft. Collins April, 2005. Title: "Education. Are there any questions?"

Lecturer, Distinguished Lecturer Series, Computer Science Dept, Purdue University, March 2005. "Are there any questions?"

Lecturer, Distinguished Lecturer Series, Rice University, Houston, TX, November, 2004. Title: Trends in Computer Architecture.

Lecturer, Distinguished Lecturer Series, Courant Institute, New York University, October, 2004. Title: The Microprocessor of the Year 2014: Do Pentium 4, Pentium M, and Power 5 provide any hints?

Lecturer, Distinguished Lecturer Series, Carnegie Mellon University, Pittsburgh, April, 2004. Title: "The Microprocessor Ten Years from now: What are the challenges, how do we meet them?"

Lecturer, City University of Hong Kong Distinguished Lecture Series on Teaching and Learning, Hong Kong, January, 2004. Title: Ten Commandments of Good Teaching.

Lecturer, Distinguished Lecture Series, University of California, Irvine, December, 2003. Title: "Fundamentals: Moore's Law, Microarchitecture, and the Microprocessor of the year 2014."

Lecturer, Worcester Polytechnic Institute Distinguished Lecturer Series, September, 2003. Title: The High Performance Microprocessor in the Year 2013: Will the laws of Physics finally catch up with the Microprocessor development cycle?"

Lecturer, The University of Texas ECEntury Distinguished Lecturer Series, March, 2003.

Robert T. Chien Memorial Lecture, University of Illinois, Urbana-Champaign, December, 2002.

Lecturer, The Erik Jonsson School of Engineering Distinguished Lecture Series, The University of Texas at Dallas, October, 2002.

Lecturer, University of Alberta Distinguished Lecturer Series in Electrical Engineering, October, 2002.

Lecturer, California State University at Fresno Distinguished Lecture Series in Computer Engineering, May, 2002.

The 2001-2002 Distinguished Lecturer in Computer Science, University of Wisconsin, La Crosse, April, 2002.

Lecturer, Georgia Tech Distinguished Lecturer Series in Computer Engineering, April, 2002.

The William Mong Distinguished Lecturer at the University of Hong Kong, March, 2002.

Lecturer, University of Illinois Distinguished Lecturer Series in Computer Science, February, 2002.

Lecturer, University of Pittsburgh Distinguished Lecturer Series, October, 2001.

The Birck Distinguished Lecturer, Purdue University Distinguished Lecture Series, March, 2001.

Lecturer, University of California, Riverside Distinguished Lecturer Series, January, 2001.

Lecturer, Colorado State University Distinguished Lecturer Series, April, 2000.

Lecturer, University of Delaware Distinguished Lecturer Series, March, 2000.

Lecturer, Texas A&M University Distinguished Lecturer Series, February, 2000.

Invited lecturer, HP2EUR Conference, Tromso, Norway, June, 1999.

Lecturer, Carnegie-Mellon University Distinguished Lecturer Series, April, 1998.

Lecturer, University of Southern California Distinguished Lecturer Series, February, 1998.

Lecturer, UCLA Distinguished Lecturer Series, November, 1996.

Lecturer, University of Southwestern Louisiana Distinguished Lecturer Series, November, 1996.

Lecturer, University of Texas, Austin Distinguished Lecturer Series, November, 1996.

Lecturer, Texas A&M University Shell Oil Distinguished Lecturer Series, February, 1996.

Distinguished lecturer, Taiwan Government. Series of three lectures, May 4,5, 1995. At ITRI, May 4: "HPS, The Microarchitecture for High Performance Processors." At Chiao Tung University, May 4: "The Next Challenge: a 10 IPC Processor." At National Taiwan University, May 5: "Branch Prediction: After Yeh's Algorithm, What?"

Invited lecturer, "Doctoral Network," a one-week summer school for leading PhD students in computer architecture, organized by the French National Education Administration, July, 1994.

Lecturer, Northwestern University Distinguished Lecturer Series, May, 1994.

Inaugural lecturer, the IEEE Distinguished Visitor Program for Asia and the Pacific. Lectures were presented in Bombay, Calcutta, Delhi, and Hong Kong, May 1993.

Lecturer, University of California, San Diego Distinguished Lecturer Series, April, 1992.

Lecturer, University of Maryland Distinguished Lecturer Series, November, 1991.

Distinguished Speaker, 1991 IEEE Joint Technical Committees Meeting, Rochester, New York, March 1991. Title: "Computer Architecture Choices.

Lecturer, Case-Western Reserve University Distinguished Lecturer Series, December, 1989.

Lecturer, Northeastern University Distinguished Lecturer Series, May, 1989.

Lecturer, Clemson University NCR Distinguished Lecturer Series, March, 1989.

At the invitation of the Ministry of Education of the Italian government, a series of 16 lectures on Advanced Concepts in Computer Architecture at the University of Pavia, May, 1985.

National Distinguished Visitor of the IEEE, 1989-93.

National ACM Lecturer: 1975-76, 1985-92, 1995-present.

INVITED PANELS at INTERNATIONAL CONFERENCES, SYMPOSIA

Member, invited panel at 40th annual International Symposium on Microarchitecture, Chicago, December, 2007, on the topic, "Computing Beyond Von Neumann."

Member, invited panel at the 13th International High Performance Computer Architecture Symposium, Phoenix, February, 2007, on the topic, "Researching Novel Systems: To Instantiate, Emulate, Simulate, or Analyticate?"

Member, invited panel at the 39th International Symposium on Microarchitecture, Orlando, Florida, December, 2006, on the topic, "The Implications of Nanotechnology."

Member, invited panel at the 12th International High Performance Computer Architecture Symposium, Austin, February, 2006, on the topic, "Patenting the Fruits of Academic Research: What are the Implications?"

Moderator, invited panel at 16th Symposium on Computer Architecture and High Performance Computing, Foz do Iguacu, Brazil, October, 2004.

Member, invited panel at 2004 IEEE International Symposium on Performance Analysis of Sytems and Software, Austin, March, 2004, on the topic, "The Future of Simulation: A Field of Dreams?"

Member, invited panel at the 10th International High Performance Computer Architecture Symposium, Madrid, February, 2004, on the topic, "Bridging the Research Gap between Academia and Industry."

Member, invited panel at the 8th International High Performance Computer Architecture Symposium, Boston, February, 2002, on the topic, "Processors, Memory, Interconnect, where will the greatest benefit come from in the next ten years."

Member, invited panel at the 34th International Symposium on Microarchitecture, Austin, TX, December, 2001, on the topic, "Intellectual Property: Microarchitecture is in the thick of it."

Member, invited panel at IEEE ICCD Symposium, Austin, September, 2001.

Member, invited panel at the 30th International Symposium on Microarchitecture, Raleigh, NC, December, 1997 on the topic, "The future of ISAs other than x86."

Member, invited panel at the 28th International Symposium on Microarchitecture, Ann Arbor, December 1995, on the topic, "Single-chip performance on the Microprocessor of the year 2000."

Moderator, one of two invited panels at the 21th International Symposium on Computer Architecture, Chicago, April 1994, on the topic, "Computer Architecture as a Discipline in the year 2000."

Moderator, one of two invited panels at the 20th International Symposium on Computer Architecture, San Diego, May 1993, on the topic, "Experimental Research: How do we Measure Success."

Member, one of two invited panels at the 19th International Symposium on Computer Architecture, Australia, May 1992, on the topic, "Processor architectures for Peta-op Performance."

Member, one of two invited panels at the 18th International Symposium on Computer Architecture, Toronto, May 1991, on the topic, "The Influence of University Research in Computer Architecture on the development of new commercial computers."

Member, one of two invited panels at the 17th International Symposium on Computer Architecture, Seattle, May 1990, on the topic, "Better than one operation per clock: Vectors, VLIW, and Superscalar."

Member, one of two invited panels at the 13th International Symposium on Computer Architecture, Tokyo, June 1986, on the topic, "AI Machines and Scientific Machines, Are they compatible?"

EDITORSHIPS

Associate Editor, *Computer Architecture Letters*, 2006-

(Founding) Editor-in-Chief, *Computer Architecture Letters*, 2001-2005.

Member, Editorial Board, *Proceedings of the IEEE*, 1996-2000.

Associate Editor of the *IEEE Transactions on Computers*, 1992-1996.

Member, Editorial Board, *IEEE Computer*, 1989-1992.

Guest Editor of a Special Issue of *IEEE Computer* (December, 1997) on the topic, Strategic Directions for Computer Architecture Research.

Guest Editor of a Special Issue of *Proceedings of the IEEE* (December, 1995) on the topic, Microprocessors.

Guest Editor of a Special Issue of *IEEE Computer* (March, 1994) on the topic, The I/O Subsystem, A Candidate for Improvement.

Guest Editor of a Special Issue of *IEEE Computer* (January, 1991) on the topic, Experimental Research in Computer Architecture and Performance Measurement.

Guest Editor of a Special Issue of *IEEE Computer* (January, 1989) on the topic, Design and Engineering Tradeoffs in the Implementation of Various Commercially Successful Computers.

EXTERNAL EXAMINING BOARDS

External Examiner, Graduate program in Electrical Engineering and Computer Engineering, University of Cincinnati, October, 2005.

External Examiner, Computer Engineering Research, National Foundation for Science and Technology, Lisbon, Portugal, January, 2005.

External Examiner, Electrical and Computer Engineering Department, Northwestern University, May, 2001.

External Examiner, Computer Science Department, UNLV, March, 2001.

External Examiner, Texas A&M PhD program in Electrical and Computer Engineering, March, 2000.

External Examiner, Computer Engineering Research, National Foundation for Science and Technology, Lisbon, Portugal, December, 1999.

External Examiner, City University of Hong Kong, Computer Science Department, 1996-2003.

Member, Dean's External Advisory Committee, School of Electrical Engineering and Computer Science, Polytechnic University of New York, 1992-1994.

PROFESSONAL SERVICE

Member of the Governing Board of the IEEE Computer Society, 1989-1993.

Vice President for Press Activities of the IEEE Computer Society, 1992.

General Co-Chairman, 12th International Symposium on High Performance Computer Architecture, Austin, TX, February, 2006.

General Chairman, 29th International IEEE/ACM Symposium on Computer Architecture (ISCA), Anchorage, May, 2002.

General Chairman, 34th International IEEE/ACM Workshop and Symposium on Microarchitecture, Austin, TX, November, 2001.

General Chairman, 21st International IEEE/ACM Workshop on Microarchitecture and Microprogramming, San Diego, CA, November, 1988.

General Chairman, First International Symposium on High Performance Computer Architecture, Raleigh, NC, January, 1995.

Program Co-Chairman, 15th International IEEE/ACM Symposium on Computer Architecture (ISCA), June, 1988.

Program Co-Chairman (Computer Architecture), IEEE/ACM International Conference on Supercomputing, Melbourne, July, 1998.

Member of the Advisory Committee, ACM Distinguished Lectureship Program, 1998-2003; Director, 2001-2002.

Chairman, Joint ACM-IEEE Computer Society Eckert-Mauchly Award Committee, 1992, 1999. Member of the committee, 1988-1992, 1998-2000.

Member, IEEE Computer Society Harry Goode Award Committee, 2001- , Chair, 2004-2006.

Member, IEEE Computer Society Wallace W. McDowell Award Committee, 2001- , Chair, 2004-2006.

Chairman of a Task Force appointed by the President-elect of the IEEE Computer Society to investigate problems dealing with the education of computer engineers and computer scientists, starting with high school preparation, continuing with educational programs in the Colleges, and finally dealing with continuing education beyond graduation, July, 1991.

Member of the Strategic Planning Committee of the IEEE Computer Society, 1991.

Member, IEEE Computer Society Taylor Booth Award Committee, 1993-1996.

Member, Educational Activities Board, IEEE Computer Society, 1988-1994.

Invited participant (one of 20), NSF Workshop on Grand Challenges in Computer Architecture for the Support of High Performance Computing, Purdue University, December 1991. A report of that workshop appeared as an invited paper (with 20 co-authors) in the Journal of Parallel and Distributed Computing. High Performance Architectures for Numerical and Symbolic Computations, University of California, Berkeley, February 13-15, 1985.

Member, Steering Committee of the 19th, 21st, 22nd, 23rd, 24th, 25th, 26th, 28th, 29th, 30th and 31st, 33rd, and 35th International Symposia on Computer Architecture, May, 1989, April, 1994, June, 1995, May, 1996, June, 1997, June, 1998, May, 1999, May, 2001, May, 2002, June, 2003, June, 2004, June, 2006, and June, 2008.

Member, Steering Committee of the 25th, 26th, 27th, 28th, 29th, 30th, 31st, 32nd, 33rd, 34th, 35th, 36th, 37th, 38th, 39th and 40th International Symposium on Microarchitecture, December, 1992, December, 1993, December, 1994, December, 1995, December, 1996, December, 1997, December, 1998, November, 1999, December, 2000, December, 2001, November, 2002, December, 2003, December, 2004, November, 2005, December, 2006, December, 2007.

Member, Steering Committee of the 2nd, 3rd, 4th, 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th, 13th, and 14th IEEE International Symposium on High Performance Computer Architecture, San Jose, February, 1996, San Antonio, February, 1997, San Francisco, February, 1998, Orlando, January, 1999, Toulouse, January, 2000, Monterrey, Mexico, January, 2001, Boston, January, 2002, Anaheim, January, 2003, Madrid, February, 2004, San Francisco, February, 2005, Austin, February, 2006, Phoenix, February, 2007, Salt Lake City, February 2008.

Member, Technical Advisory Board, IEEE Technical Committee on Computer Architecture, 1995-present.

Organizer/co-ordinator, minitrack on the subject: "Real Machines: Design Choices, Engineering Trade-offs," at the 20th Hawaii International Conference on Systems Science, January, 1989.

Organizer/co-ordinator, minitrack on the subject: "Experimental Research in Computer Architecture," at the 21st Hawaii International Conference on Systems Science, January, 1990.

Organizer/co-ordinator, second consecutive minitrack on the subject: "Experimental Research in Computer Architecture," at the 22nd Hawaii International Conference on Systems Science, January, 1991.

Organizer/co-ordinator, minitrack on the subject: "I/O Architecture: Issues and Bottlenecks," at the 23rd Hawaii International Conference on Systems Science, January, 1992.

Organizer/co-ordinator, second consecutive minitrack on the subject: "I/O Architecture: Issues and Bottlenecks," at the 24th Hawaii International Conference on Systems Science, January, 1993.

Lecturer, IEEE Computer Society's Tutorial Program on the subject, "Computer Architecture Choices."

Lecturer, IEEE Computer Society's one-day tutorial for High School Math and Science teachers, 1991-94.

External reviewer, Research Programs Review at the Lawrence Livermore National Laboratory, December, 1984.

Member, organizing committee for the First Northern California Universities Supercomputer Workshop, Lawrence Livermore National Laboratory, February, 1984.

Member, Program Committee for more than 50 conferences and symposia, including most recently the 31th International Workshop on Microarchitecture and Microprogramming (Dallas, December 1998), the 1998 3rd International conference on Parallel Architectures and Compiling Techniques (Paris, October, 1998), the 5th International High Performance Computer Architecture Conference (Orlando, January, 1999), 32th International Symposium on Microarchitecture (Haifa, November, 1999), HPCA-7 (Monterrey, January, 2001), ISCA-2001 (Goteborg, June, 2001), 8th International Conference on High Performance Computing (Hyderabad, December, 2001), and ICS (New York City, June, 2002), ISCA-2004 (Munich, June 2004), Micro-37 (San Francisco, Dec 2004), HiPEAC (Barcelona, Nov 2005), Micro-38 (Barcelona, Nov 2005), ISCA (Boston, 2006), Micro-39 (Orlando, 2006), HPCA-13 (Phoenix, 2007), Micro-40 (Chicago, 2007), HPCA-14 (Salt Lake City, 2008), ISCA (Beijing, 2008), SBAC-PAD (Brazil, 2008).

Reviewer, NSF proposals, ACM and IEEE journals and conferences.

Accreditation visitor, ABET (computer engineering) 1990-1994.

Accreditation visitor, CSAB (computer science), 1986-1992.

Chairman, Curriculum Assistance Committee of the IEEE Computer Society, 1978-1981. In this capacity I organized and conducted workshops dealing with the Model Curriculum in Computer science and Engineering in 1979 (San Francisco), 1980 (Chicago) and 1981 (Washington, D.C.).

General Chairman, 5th Annual Southeastern Symposium on System Theory, Co-hosted by N.C. State, UNC-Chapel Hill, Duke, March, 1973.

Member, Steering Committee, COINS-3, 1969.

Program Chairman, 1975 Southeastern Regional ACM Meeting, April, 1975.

Session Chairman, more than 50 conferences, including: COINS-3 (January, 1969), Hawaii International Conference on Systems Sciences (January, 1970), 6th Annual Southeastern Systems Theory Symposium (1974), 4th International Multi-valued Logic Symposium (1974), 7th Annual Southeastern Systems Theory Symposium (1975), Fall Compcon (September, 1981), Spring Compcon (February, 1983), Spring Compcon (February, 1984), Micro-17 (October, 1984), Spring Compcon (February, 1985), Micro-18 (December, 1985), HICSS-19 (January, 1986), Spring Compcon (March, 1986), 13th ICAC (June, 1986), Micro-19 (October, 1986), Micro-20 (December, 1987), Compcon 88 (March, 1988), Micro-21 (December, 1988), Compcon 89 (February, 1989), International Supercomputing Conference (June, 1989), HICSS-1991 (January, 1991), Micro-24 (November, 1991), ISCA-13 (June, 1986), HICSS-1992 (January, 1992), ISCA-19 (May, 1992), Micro-25 (November, 1992), IFIP Workshop on Architecture and Compiling (January, 1993), HICSS-1993 (January, 1993), Micro-26 (December, 1993), IPPS (April, 1994), Micro-27 (December, 1994), HPCA-1 (January, 1995), ISCA (June, 1995), PACT (June, 1995), Micro-28 (December, 1995), ICS (July, 1997), Micro-30 (December, 1997), HPCA-5 (January, 1999), PACT (October, 1999), ISCA (June, 2004), ISCA (June, 2006), HPCA (February, 2008).

TUTORIALS (Sponsored by the professional computer societies, IEEE_CS and ACM)

Instructor, one day tutorial on current hot topics in Computer Architecture, as follows: IEEE Tutorial Week West, October 1, 1984, IEEE Tutorial Week West, June 7, 1985, IEEE Compcon 86, March 1, 1986, Compcon 88, February 29, 1988, Compcon 89, February 27, 1989, Compcon 90, February 26, 1990, Compcon 91, February 25, 1991, Compcon 92, February 24, 1992, Compcon 93, February 22, 1993, Compcon 94, February 28, 1994, Compcon 95, March 5, 1995, Supercomputing 95, December 4, 1995, Compcon 96, February 25, 1996, Compcon 97, February 23, 1997, Supercomputing 2001, November 11, 2001.

Instructor, half-day tutorial on architectural mechanisms for exploiting concurrency, as follows: 17th IEEE/ACM International Symposium on Computer Architecture, May 28, 1990, 18th IEEE/ACM International Symposium on Computer Architecture, May 27, 1991, 19th IEEE/ACM International Symposium on Computer Architecture, May 17, 1992, International Conference on Parallel Processing, August 17, 1992, 21st IEEE/ACM International

Symposium on Computer Architecture, April 19, 1994, 1994 International Conference on Parallel Architectures and Compiler Technology, August 23, 1994, 22nd IEEE/ACM International Symposium on Computer Architecture, June 21, 1995, the 1995 International Conference on Parallel Architectures and Compiler Techniques, June 27, 1995, the 23rd IEEE/ACM International Symposium on Computer Architecture, May 21, 1996, IEEE Hot Chips Symposium, August 18, 1996, and ICCD, October 7, 1996.

Instructor, one-day tutorial to non-Computer-Architects on the state-of-the-art in Computer Architecture, as follows: 26th Annual HICSS Conference, January 5, 1993, 27th Annual HICSS Conference, January 4, 1994, 28th Annual HICSS Conference, January 3, 1995, 29th Annual HICSS Conference, January 2, 1996.

Instructor, one-day tutorial on Microarchitecture: Concepts, Tradeoffs, the Future, PACT 2007, Brasov, Romania, September, 2007.

BOOKS

Introduction to Computing Systems: From Bits and Gates to C and Beyond, McGraw-Hill, 2001 (ISBN No. 0-07-237690-2), co-authored with Professor Sanjay Jeram Patel. 2nd edition, McGraw-Hill, 2004 (ISBN No. 0-07-246750-9, ISBN 0-07-121503-4(ISE)).

MAGAZINE/NEWSLETTER ARTICLES,TV/RADIO INTERVIEWS, BANQUET TALKS

Invited speaker, "eContent Summit: Scenarios for the future," a workshop organized by the President of Croatia, Brijuni, September, 2007. Title: eContent: Static vs. Dynamic; In Education, it makes a huge difference.

Featured speaker, IEEE Computer Society 60th Anniversary Celebration, Hotel Intercontinental, San Juan, Puerto Rico, June, 2006. Title: The Future of Computer * (Are we in Serious Trouble?).

Banquet address, 19th IEEE International Parallel & Distributed Processing Symposium, Denver, April, 2005. Title: A Unifying Theory of Distributed Processing.

Adaptive Insertion Policies for High-Performance Caching. *IEEE Micro,* Top Picks Special Issue, January/February, 2008. [earlier version in *Proceedings, 34th annual IEEE/ACM International Symposium on Computer Architecture,* San Diego, CA, June 2008 (with Moinuddin K. Qureshi, Aamer Jaleel, Simon C. Steely Jr., and Joel Emer).

Diverge-Merge Processor: Generalized and Energy-Efficient Dynamic Predication, *IEEE Micro,* Top Picks Special Issue, January/February, 2007. [earlier version in *Proceedings, 39th annual IEEE/ACM International Symposium on Microarchitecture,* Orlando, Florida, December, 2006] (with Hyesoon Kim, Jose Joao, and Onur Mutlu)..XP Techniques for Efficient Processing in Runahead Execution Engines. *IEEE Micro,* Top Picks Special Issue, January/February, 2006. [earlier version in *Proceedings, 32nd annual IEEE/ACM International Symposium on Computer Architecture,* Madison, Wisconsin, June, 2005] (with Onur Mutlu and Hyesoon Kim).

Wish Branches: Combining Conditional Branching and Predication for Adaptive Predicated Execution, *IEEE Micro,* Top Picks Special Issue, January/February, 2006. [earlier version in *Proceedings, 38th annual IEEE/ACM International Symposium on Microarchitecture,* Barcelona, November, 2005] (with Hyesoon Kim, Onur Mutlu, and Jared Stark).

Runahead Execution: An Effective Alternative to Large Instruction Windows, *IEEE Micro,* Top Picks Special Issue, November/December, 2003. [earlier version in *Proceedings, 9th annual IEEE High Performance Computer Architecture Symposium.* Anaheim, CA, February, 2003] (with Onur Mutlu, Jared Stark, and Chris Wilkerson).

Interview on "Newsmakers," WLSU Radio, La Crosse Wisconsin, on the future of computer technology. Aired May 6, 2002.

Interview on "Generation.e," CNBC Asia, on the topic, "The future capabilities of microprocessors," March 8, 2002.

"New Requirements, Bottlenecks, and Good Fortune, Agents of Evolution of the Microprocessor," *Proceedings of the IEEE,* November, 2001.

"The First Computing Course for CS, CE, and EE Majors at Michigan," *The Interface* (a Newsletter published jointly by ASEE and the IEEE Education Society), November 1998, pp. 1-3.

"Identifying Obstacles in the Path to *More*," *IEEE Computer,* December 1997, Vol. 30, No. 12, p. 32.

"One billion transistors, one uniprocessor, one chip," *IEEE Computer,* September, 1997.

"First, Let's Get the Uniprocessor Right," *Microprocessor Report,* (Invited article for 25th Anniversary of the Microprocessor Special Issue), August 5, 1996, pp. 23-24.

"The Microprocessor for Scientific Computing in the Year 2000," *IEEE Computational Science & Engineering,* Summer 1996, pp. 42-43.

"Education in Computer Science and Computer Engineering Starts with Computer Architecture," *Computer Architecture Technical Committee Newsletter,* June 1996.

"Scanning the Special Issue on Microprocessors," *Proceedings of the IEEE,* December 1995, Vol. 83, No. 12.

"The I/O Subsystem - A Candidate for Improvement," *IEEE Computer,* March 1994, Vol. 27, No. 3, pp. 15-16.

"Disk Arrays: High-Performance, High-Reliability Storage Subsystems," *IEEE Computer,* March 1994, Vol. 27, No. 3, pp. 30-36.

"Experimental Research in Computer Architecture," *IEEE Computer,* January 1991, Vol. 24, No. 1, pp. 14-16.

"Real Machines: Design Choices/Engineering Trade-Offs," *IEEE Computer,* January 1989, Vol. 22, No. 1, pp. 8-10.

RESEARCH PUBLICATIONS

Archival Journals:

Dynamic Predication of Indirect Jumps, *IEEE Computer Architecture Letters,* Vol. 6, May 2007 (with Jose A. Joao, Onur Mutlu, and Hyesoon Kim).

Address-Value Delta (AVD) Prediction: A Hardware Technique for Efficiently Parallelizing Dependent Cache Misses, *IEEE Transacctions on Computers,* vol. 55, no. 12, December, 2006, (with Onur Mutlu and Hyesoon Kim).

An Analysis of the Performance Impact of Wrong-Path Memory References on Out-of-Order and Runahead Execution Processors, *IEEE Transactions on Computers,* vol. 54, no. 12, December, 2005, (with Onur Mutlu, Hyesoon Kim, and David Armstrong).

Using the First-Level Caches as Filters to Reduce the Pollution Caused by Speculative Memory References, *International Journal of Parallel Programming,* 2005, (with Onur Mutlu, Hyesoon Kim, and David N. Armstrong).

On Reusing the Results of Pre-Executed Instructions in a Runahead Execution Processor, *Computer Architecture Letters,* vol 4, no. 1, Jan. 2005 (with O. Mutlu, H. Kim, and J. Stark).

Soft Updates: A Solution to the Metadata Update Problem in File Systems, *ACM Transactions on Computer Systems,* vol. 18, no. 2, May 2000, pp 127-153 (with Greg Ganger, Marshall Kirk McKusick, and Craig A.N. Soules).

Evaluation of Design Options for the Trace Cache Fetch Mechanism, *IEEE Transactions on Computers,* February, 1999, (with Sanjay Patel and Daniel Friendly).

Increasing the Instruction Fetch Rate via Block-structured Instruction Set Architectures, *International Journal of Parallel Programming,* vol.26, no.4, August, 1998 (with E. Hao, P-Y Chang, M. Evers).

Using System Level Models to Evaluate I/O Subsystem Designs, *IEEE Transactions on Computers,* vol 47, no. 6 (June, 1998) (with Gregory Ganger).

Improving Branch Prediction Accuracy by Reducing Pattern History Table Interference, *International Journal of Parallel Programming,* 1997, vol 25, num 5 (with P-Y Chang and Marius Evers).

The Effects of Mispredicted-Path Execution on Branch Prediction Structures, *International Journal of Parallel Programming,* 1997, vol 25, num 5, pp. 363-384 (with Stephan Jourdan, Tse-Hao Hsing and Jared Stark).

Using Predicated Execution to Improve the Performance of a Dynamically Scheduled Machine with Speculative Execution, *International Journal of Parallel Programming,* vol.24, 1996 (with P-Y Chang, E. Hao, and P.

Chang).

Branch Classification: A New Mechanism for Improving Branch Predictor Performance, *International Journal of Parallel Programming,* vol.24, 1996 (with Po-Yung Chang, Eric Hao, and Tse-Yu Yeh).

Enhancing Instruction Scheduling With a Block-Structured ISA, *International Journal of Parallel Programming,* vol. 23, no. 3, 1995 (with Stephen Melvin).

Report of the Purdue Workshop on Grand Challenges in Computer Architecture for the Support of High Performance Computing, *Journal of Parallel and Distributed Computing,* vol.16, pp. 199-211, 1992 (with 19 co-authors).

An Experimental Single-Chip Data Flow CPU, *IEEE Journal of Solid State Circuits,* January, 1992 (with G.Uvieghara, W. Hwu, Y. Nakagome, D. Jeong, D. Lee, D. Hodges).

Alternative Implementations of Prolog, the Microarchitecture Perspective, *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 19, no. 4, pp. 693-698, July/August, 1989.

Checkpoint Repair for High Performance Out-of-Order Execution Machines, *IEEE Transactions on Computers,* December, 1987, (with Wen-mei Hwu).

Retrofitting the VAX-11/780 Microarchitecture for IEEE Floating Point Arithmetic - Implementation Issues, Measurements, and Analysis, *IEEE Transactions on Computers* v. c-34, no. 8, August, 1985 (with David Aspinwall).

Independent necessary conditions for functional completeness in m-valued logic, *Notre Dame Journal of Formal Logic,* v. 18, no. 2, April, 1977.

Some clarifications of the concept of a Garden of Eden configuration, *Journal of Computer and System Sciences,* v. 10, no. 1, February, 1975 (with S. Amoroso and G. Cooper).

Optimal and near optimal universal logic modules with interconnected external terminals, *IEEE Transactions on Computers,* October, 1973.

Minimum search tree structures for data partitioned into pages, *IEEE Transactions on Computers,* September, 1972.

Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures, *Journal of Computer and Systems Sciences,* October, 1972 (with S. Amoroso).

Variable length file structures having minimum average search time, *Communications of the ACM,* v. 12, no. 2, February, 1969.

Refereed Conferences and Symposia:

Achieving Out-of-Order Performance with Almost In-Order Complexity, *Proceedings, 35th International Symposium on Computer Architecture,* Beijing, China, June 2008, (with Francis Tseng).

Feedback-Driven Threading: Power-Efficient and High-Performance Execution of Multi-threaded Workloads on CMP, *Proceedings of ASPLOS XIII,* Seattle, March 2008 (with M. Aater Suleman and Moinuddin Qureshi).

Improving the Performance of Object-Oriented Languages with Dynamic Predication of Indirect Jumps, *Proceedings of ASPLOS XIII,* Seattle, March 2008 (with Jose A. Joao, Onur Mutlu, and Hyesoon Kim).

A Performance-aware Speculation Control Technique using Wrong Path Usefulness Prediction, *Proceedings of the 14th Annual IEEE High Performance Computer Architecture Symposium (HPCA-14),* Salt Lake City, February 2008 (with Chang Joo Lee, Hyesoon Kim, and Onur Mutlu).

Adaptive Insertion Policies for High Performance Caching, *Proceedings, 34th International Symposium on Computer Architecture,* San Diego, CA, June 2007 (with Moinuddin Qureshi, Aamer Jaleel, Simon Steely, Jr. and Joel Emer).

VPC Prediction: Reducing the Cost of Indirect Branches via Hardware-Based Dynamic Devirtualization, *Proceedings, 34th International Symposium on Computer Architecture,* San Diego, CA, June 2007 (with Hyesoon Kim, Jose Joao, and Onur Mutlu).

Profiling-assisted Compiler Support for Dynamic Predication in Diverge-Merge Processors, *5th International Symposium on Code Generation and Optimization (CGO-5),* San Jose, March 2007 (with Hyesoon Kim, Jose

Joao, and Onur Mutlu).

Line Distillation: Increasing Cache Capacity by Filtering Unused Words in Cache Lines, *Proceedings of the 13th Annual IEEE High Performance Computer Architecture Symposium (HPCA-13),* Phoenix, February, 2007 (with Moinuddin Qureshi and Muhammed Aater Suleman).

Feedback Directed Prefetching: Improving the Performance and Bandwidth-Efficiency of Hardware Prefetchers, *Proceedings of the 13th Annual IEEE High Performance Computer Architecture Symposium (HPCA-13),* Phoenix, February, 2007 (with Santhosh Srinath, Onur Mutlu, and Hyesoon Kim).

Diverge-Merge Processor(DMP):Dynamic Predicated Execution of Complex Control-Flow Graphs Based on Frequently Executed Paths *Proceedings, 39th annual IEEE/ACM International Symposium on Microarchitecture,* Orlando, December, 2006 (with Hyesoon Kim, Jose A. Joao, and Onur Multu).

Utility-Based Cache Partitioning: A Low-Overhead, High-Performance, Runtime Mechanism to Partition Shared Caches, *Proceedings, 39th annual IEEE/ACM International Symposium on Microarchitecture,* Orlando, December, 2006 (with Moinuddin Qureshi).

A Case for MLP-Aware Cache Replacement, *Proceedings, 33rd International Symposium on Computer Architecture,* Boston, June, 2006 (with Moinuddin Qureshi, Daniel Lynch, and Onur Mutlu).

2D-Profiling: Detecting Input-Dependent Branches with a Single Input Data Set, *4th International Symposium on Code Generation and Optimization (CGO-4),* New York City, March, 2006 (with Hyesoon Kim, Mohammed Aater Suleman, and Onur Mutlu).

Wish Branches: Combining Conditional Branching and Predication for Adaptive Predicated Execution, *Proceedings, 38th annual IEEE/ACM International Symposium on Microarchitecture,* Barcelona, November, 2005 (with Hyesoon Kim, Onur Mutlu, and Jared Stark).

Address-Value Delta (AVD) Prediction: Increasing the Effectiveness of Runahead Execution by Exploiting Regular Memory Allocation Patterns, *Proceedings, 38th annual IEEE/ACM International Symposium on Microarchitecture,* Barcelona, November, 2005 (with Onur Mutlu and Hyesoon Kim).

The V-Way Cache: Demand Based Associativity via Global Replacement, *Proceedings, 32nd International Symposium on Computer Architecture,* Madison, Wisconsin, June, 2005 (with Moinuddin Qureshi and David Thompson).

Techniques for Efficient Processing in Runahead Execution Engines. *Proceedings, 32nd International Symposium on Computer Architecture,* Madison, Wisconsin, June, 2005 (with Onur Mutlu and Hyesoon Kim).

Microarchitecture-Based-Introspection: A Technique for Transient-Fault= Tolerance in Microprocessors, *The IEEE International Conference on Dependable Systems and Networks (DSN 2005),* Yokahama, Japan, June 2005 (with Moinuddin Qureshi and Onur Mutlu).

Early Detection of Branch Mispredictions using Wrong Path Execution Information, *Proceedings, 37th annual IEEE/ACM International Symposium on Microarchitecture,* Portland, OR, December, 2004 (with Onur Mutlu, Hyesoon Kim, and David Armstrong).

Cache Filtering Techniques to Reduce the Negative Impact of Useless Speculative Memory References on Processor Performance, *Proceedings of the 16th Symposium on Computer Architecture and High Performance Computing,* Foz do Iguacu, Brazil, October, 2004, (with Onur Mutlu, Hyesoon Kim, and David Armstrong).

Partitioned First-level Cache design for Clustered Microarchitectures, *Proceedings, 17th annual ACM International Conference on Supercomputing,* San Francisco, June, 2003 (with P. Racunas).

Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors, *Proceedings, 9th Annual IEEE International Symposium on High Performance Computer Architecture,* Anaheim, CA, February, 2003 (with Onur Mutlu, Jared Stark, Chris Wilkerson).

Microarchitectural Support for Precomputation Microthreads, *Proceedings of the 35th International Symposium on Microarchitecture,* Istanbul, November, 2002 (with R. Chappell, F. Tseng, and A. Yoav).

Handling of Packet Dependencies: A Critical Issue for Highly Parallel Network Processors, *Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES 2002),* Grenoble, France, October, 2002 (with S. Melvin).

Difficult-Path Branch Prediction Using Subordinate Microthreads, *Proceedings, 29th International Symposium on Computer Architecture,* Anchorage, May, 2002 (with R. Chappell).

Pipelined and Redundant Binary Adders for Modern Processors, *Proceedings, 8th Annual IEEE International Symposium on High Performance Computer Architecture,* Boston, MA, February, 2002 (with M. Brown).

Select-Free Instruction Scheduling Logic, *Proceedings, 34th International Symposium on Microarchtecture,* Austin, TX, December, 2001 (with M. Brown and J. Stark).

On Pipelining Dynamic Instruction Scheduling Logic, *Proceedings, 33rd International Symposium on Microarchtecture,* Monterey, CA, December, 2000 (with J. Stark and M. Brown).

Simultaneous Subordinate Microthreading (SSMT), *Proceedings, 26th International Symposium on Computer Architecture,* Atlanta, GA, May, 1999 (with R. Chappell, J. Stark, S.W.P. Kim, and S. Reinhardt).

Putting the Fill Unit to Work: Dynamic Optimizations for Trace Cache Microprocessors, *Proceedings, 31st International Symposium on Microarchitecture,* Dallas, TX, November, 1998 (with Daniel Friendly and Sanjay Patel).

Variable Length Path Branch Prediction, *Proceedings, ASPLOS VIII,* San Jose, CA, October, 1998 (with Jared Stark and Marius Evers).

An analysis of correlation and predictability: What makes two-level branch predictors work, *Proceedings, 25th International Symposium on Computer Architecture,* Barcelona, June, 1998 (with Marius Evers, Sanjay J Patel, and Robert S Chappell).

Improving Trace Cache Effectiveness with Branch Promotion and Trace Packing, *Proceedings, 25th International Symposium on Computer Architecture,* Barcelona, June, 1998 (with Sanjay J Patel and Marius Evers).

Alternative Fetch and Issue Policies for the Trace Cache Fetch Mechanism, *Proceedings, 30th International Symposium on Microarchitecture,* Raleigh, NC, December, 1997 (with Daniel Friendly and Sanjay Patel).

Reducing the Performance Impact of Instruction Cache Misses by Writing Instructions into the Reservation Stations Out-of-order, *Proceedings, 30th International Symposium on Microarchitecture,* Raleigh, NC, December, 1997 (with Jared Stark and Paul Racunas).

Using Non-Volatile Storage to Improve the Reliability of RAID5 Disk Arrays, *Proceedings, International Symposium on Fault Tolerant Computer Systems,* Seattle, June, 1997 (with Robert Hou).

Target Prediction for Indirect Jumps, *Proceedings of the 24th International Symposium on Computer Architecture,* Denver, June 1997 (with P-Y. Chang).

The Agree Predictor: A Mechanism for Reducing Negative Branch History Interference, *Proceedings of the 24th International Symposium on Computer Architecture,* Denver, June 1997 (R. Chappell, E. Sprangle, and M. Alsup).

Increasing the Instruction Fetch Rate via Block-Structured Instruction Set Architectures, *Proceedings, 29th International Symposium on Microarchitecture,* Paris, France, December, 1996 (with Eric Hao, Po-Yung Chang, and Marius Evers).

Improving Branch Prediction Accuracy by Reducing Pattern History Table Interference, *Proceedings, International Conference on Parallel Architectures and Compilation Techniques,* Boston, October 1996 (with P-Y Chang and M. Evers)

The Effects of Mispredicted-Path Execution on Branch Prediction Structures, *Proceedings, International Conference on Parallel Architectures and Compilation Techniques,* Boston, October 1996 (with S Jourdan, T-H Hsing and J Stark).

Using Hybrid Branch Predictors to Improve Branch Prediction Accuracy in the Presence of Context Switches, *Proceedings of the 23rd International Symposium on Computer Architecture,* Philadelphia, May, 1996 (with Marius Evers, Po-Yung Chang, and Tse-Yu Yeh).

Alternative Implementations of Hybrid Branch Predictors, *Proceedings of the 28th International Symposium on Microarchitecture,* Ann Arbor, Michigan, November, 1995 (with Po-Yung Chang and Eric Hao).

Track Piggybacking: An Improved Rebuild Algorithm for RAIDs Disk Arrays, *Proceedings of International Conference on Parallel Processing,* August, 1995 (with Robert Hou).

Using Predicated Execution to Improve the Performance of a Dynamically Scheduled Machine with Speculative Execution, *Proceedings, International Conference on Parallel Architectures and Compilation Techniques,* Limassol, Cyprus, June 1995 (with P-Y Chang, E. Hao, and P. Chang).

On-Line Extraction of SCSI Disk Drive Parameters *Proceedings of the 1995 ACM SIGMETRICS Conference,* May, 1995 (with Gregory Ganger, Bruce Worthington, and John Wilkes).

Branch Classification: A New Mechanism for Improving Branch Predictor Performance, *Proceedings of the 27th International Symposium on Microarchitecture,* San Jose, California, November, 1994 (with Po-Yung Chang, Eric Hao, and Tse-Yu Yeh).

Facilitating Superscalar Processing via a Combined Static/Dynamic Register Renaming Scheme, *Proceedings of the 27th International Symposium on Microarchitecture,* San Jose, California, November, 1994 (with Eric Sprangle).

The Effect of Speculatively Updating Branch History on Branch Prediction Accuracy, Revisited, *Proceedings of the 27th International Symposium on Microarchitecture,* San Jose, California, November, 1994 (with Eric Hao).

Metadata Update Performance in File Systems, *Proceedings of the First Operating Systems Design and Implementation Symposium,* Monterey, November 1994 (with Gregory Ganger).

Scheduling Algorithms for Modern Disk Drives *Proceedings of the 1994 ACM SIGMETRICS Conference,* Nashville, TN, May, 1994 (with Gregory Ganger and Bruce Worthington).

Branch History Table Indexing to Prevent Pipeline Bubbles in Wide-Issue Superscalar Processors, *Proceedings of the 26th International Symposium and workshop on Microarchitecture,* Austin, TX, December 1993 (with Tse-Yu Yeh).

A Comparative Performance Evaluation of Various State maintenance Mechanisms, *Proceedings of the 26th International Symposium and workshop on Microarchitecture,* Austin, TX, December 1993 (with Michael Butler).

Trading Disk Capacity for Performance *Proceedings of the 2nd International Symposium on High Performance Distributed Computing,* Spokane, WA, July, 1993 (with Robert Hou)

Increasing Instruction Fetch Rate via Multiple Branch Prediction and a Branch Address Cache. *Proceedings of the 7th ACM International Conference on Supercomputing.* Tokyo, Japan, July, 1993. (with Tse-Yu Yeh and Deborah Marr)

A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History. *Proceedings, 20th International Symposium on Computer Architecture,* San Diego, CA, May, 1993. (with Tse-Yu Yeh)

The Process Flow Model: Examining I/O Performance from the System's Point of View, *Proceedings of the 1993 ACM SIGMETRICS Conference,* Santa Clara, CA, May, 1993 (with Gregory Ganger)

Comparing Rebuild Algorithms for Mirrored and RAID5 Disk Arrays *Proceedings, SigMOD Conference '93* Washington, DC, May, 1993 (with Robert Hou)

Disk Subsystem Load Balancing: Disk Striping vs. Conventional Data Placement *Proceedings, 26th Hawaii International Conference on Systems Sciences* Maui, January, 1993. (with Gregory R. Ganger, Bruce L. Worthington, and Robert Y. Hou)

Balancing I/O Response Time and Disk Rebuild Time in a RAID 5 Disk Array *Proceedings, 26th Hawaii International Conference on Systems Sciences* Maui, January, 1993. (with Robert Y. Hou and Jai Menon)

An Investigation of the Performance of Various Dynamic Scheduling Techniques, *Proceedings, 25th International Symposium and workshop on Microarchitecture,* Portland OR, November, 1992. (with Michael Butler)

A Comprehensive Instruction Fetch Mechanism for a Processor Supporting Speculative Execution, *Proceedings, 25th International Symposium and workshop on Microarchitecture,* Portland OR, November, 1992. (with Tse-Yu Yeh)

Alternative Implementations of Two-Level Adaptive Training Branch Prediction, *Proceedings, 19th International Symposium on Computer Architecture,* Queensland, Australia, May, 1992. (with Tse-Yu Yeh)

Toward the Specification of an ISA for High Performance Computing Engines -- Part I: The Hardware Perspective. *Proceedings, 25th Hawaii International Conference on Systems Sciences* Kauai, January, 1992. (with Michael Butler and David Dyer)

Issues and Problems in the I/O Subsystem, Part I -- The Magnetic Disk, *Proceedings, 25th Hawaii International Conference on Systems Sciences* Kauai, January, 1992. (with Robert Hou, Gregory Ganger and Charles Gimarc)

Two-Level Adaptive Branch Prediction, *Proceedings, 24th International Symposium and workshop on Microarchitecture,* Albuquerque, November, 1991. (with Tse-Yu Yeh)

The Effect of Real Data Cache Behavior on the Performance of a Microarchitecture that Supports Dynamic Scheduling, *Proceedings, 24th International Symposium and workshop on Microarchitecture,* Albuquerque, November, 1991. (with Michael Butler)

Single Instruction Stream Parallelism is Greater than Two, *Proceedings of the 18th International Symposium on Computer Architecture,* May, 1991, (with M. Butler, T-Y Yeh, M. Alsup, H. Scales, and M. Shebanow).

Exploiting Fine-Grained Parallelism through Combined Hardware and Software Techniques, *Proceedings of the 18th International Symposium on Computer Architecture,* May, 1991, (with Stephen Melvin).

An Area-efficient Register Alias Table for Implementing HPS, *Proceedings of the International Conference on Parallel Processing,* August, 1990, (with Michael Butler).

An Experimental Single-Chip Data Flow CPU *Proceedings of the 1990 Symposium on VLSI Circuits* Honolulu, June, 1990 (with G. Uvieghara, W. Hwu, et.al.).

Methodologies for Experimental Research in Computer Architecture and Performance Measurement, *Proceedings of the 23rd Annual Hawaii International Conference on System Sciences,* Kona, HI, January, 1990.

Unification Parallelism: How much can we exploit, *Proceedings of the North American Conference on Logic Programming,* Cleveland, OH, October, 1989 (with A. Singhal).

Alternative Microarchitecture Structures for Implementing the VAX Architecture, *Proceedings of the 22nd International Workshop on Microarchitecture and Microprogramming,* Dublin, August, 1989.

Performance Benefits of Large Execution Atomic Units in Dynamically Scheduled Machines, *Proceedings of the 3rd International Conference on Supercomputing,* Crete, June, 1989 (with S. Melvin).

A High Performance Prolog Processor with Multiple Function Units, *Proceedings of the 16th International Symposium on Computer Architecture,* Jerusalem, May, 1989 (with A. Singhal).

Tailoring Functional Units and Memory in a High Performance Prolog Architecture, *Proceedings of the 22nd Hawaii International Conference on Systems Sciences,* January, 1989 (with A. Singhal).

Extending a Prolog Architecture for High Performance Numeric Computations, *Proceedings of the 22nd Hawaii International Conference on Systems Sciences,* January, 1989 (with R. Yung and A. Despain).

An Extended Prolog Architecture for Integrated Symbolic and Numeric Execution, *Proceedings of the International Computer Science Conference,* December 1988 (with Robert Yung, Alvin M. Despain).

Implementing a Prolog Machine with Multiple Functional Units, *Proceedings of the 21st Workshop on Microarchitecture and Microprogramming,* November, 1988 (with A. Singhal).

Hardware Support for Large Atomic Units in Dynamically Scheduled Machines, *Proceedings of the 21st Workshop on Microarchitecture and Microprogramming,* November, 1988 (with S. Melvin and M. Shebanow).

Hierarchical Registers for Scientific Computers, *Proceedings of the ACM International Conference on Supercomputing,* Saint Malo, France, July, 1988 (with J. Swensen).

The Use of Microcode Instrumentation for Development, Debugging, and Tuning of Operating System Kernels, *Proceedings of the 1988 ACM SIGMETRICS Conference,* Santa Fe, NM, May, 1988 (with S. Melvin).

HPSm2: A Refined Single Chip Microengine, *Proceedings of the 21st Annual Hawaii International Conference on Systems Sciences,* Kona, HI, January, 1988 (with W.Hwu).

The DSI and Below: The Architecture/Hardware Component of a Computer Science Curriculum, *Proceedings of the 21st Annual Hawaii International Conference on Systems Sciences,* Kona, HI, January, 1988.

SPAM: A Microcode-Based Tool for Tracing Operating System Performance, *Proceedings, 20th Annual Workshop on Microprogramming,* Colorado Springs, CO, December, 1987 (with S. Melvin).

Exploiting Horizontal and Vertical Concurrency via the HPSm Microprocessor, *Proceedings, 20th Annual Workshop on Microprogramming,* Colorado Springs, CO, December, 1987 (with W. Hwu).

On Tuning the Microarchitecture of an HPS Implementation of the VAX *Proceedings, 20th Annual Workshop on Microprogramming,* Colorado Springs, CO, December, 1987 (with J. Wilson, S. Melvin, et.al.).

A CMOS Chip for a Prolog Processor, *Proceedings of ICCD,* New York, October, 1987 (with V. Srini et. al.).

Checkpoint Repair for Out-of-order Execution Machines, *Proceedings, 14th Annual International Symposium on Computer Architecture,* Pittsburgh, Pennsylvania, June 1987 (with W. Hwu).

Fast Temporary Storage for Serial and Parallel Computation, *Proceedings, 14th Annual International Symposium on Computer Architecture,* Pittsburgh, Pennsylvania, June 1987 (with J. Swensen).

Advantages of Implementing Prolog by Microprogramming a Host General Purpose Computer *Proceedings, 4th International Conference on Logic Programming,* Parkville, Victoria, Australia, May 1987 (with Jeff Gee and Stephen Melvin).

VLSI Implementation of a Prolog Processor *Proceedings, Stanford VLSI Conference,* Stanford, California, March 1987 (with V. Srini, et. al.).

Design Choices for the HPSm Microprocessor Chip, *Proceedings, 20th Annual Hawaii International Conference on Systems Sciences,* Kona, Hawaii, January 1987 (with W. Hwu).

A Clarification of the Dynamic/Static Interface, *Proceedings, 20th Annual Hawaii International Conference on Systems Sciences,* Kona, Hawaii, January 1987 (with S. Melvin).

Run-Time Generation of HPS Microinstructions from a VAX Instruction Stream, *Proceedings, 19th Annual International Workshop on Microprogramming,* New York City, October, 1986 (with S. Melvin, W. Hwu, M. Shebanow, C. Chen, and J. Wei).

A Microcode-Based Environment for Non-Invasive Performance Analysis, *Proceedings, 19th Annual International Workshop on Microprogramming,* New York City, October, 1986 (with S. Melvin).

The Implementation of Prolog via VAX 8600 Microcode *Proceedings, 19th Annual International Workshop on Microprogramming,* New York City, October, 1986 (with J. Gee and S. Melvin).

Several Implementations of Prolog, the Microarchitecture Perspective, *1986 IEEE International Conference on Systems, Man, and Cybernetics,* Atlanta, October, 1986.

HPSm, a High Performance Restricted Data Flow Architecuture Having Minimal Functionality, *Proceedings, 13th Annual International Symposium on Computer Architecture,* Tokyo, June, 1986 (with Wen-mei Hwu).

High Performance Prolog, The Multiplicative Effect of Several Levels of Implementation *Proceedings, Compcon86* San Francisco, CA, March, 1986 (with A.M. Despain)

Experiments with HPS, a Restricted Data Flow Microarchitecture for High Performance Computers *Proceedings, Compcon86* San Francisco, CA, March, 1986

An HPS Implementation of VAX; Initial Design and Analysis *Proceedings of the Hawaii International Conference on Systems Sciences* Honolulu, HI, January 1986 (with W.Hwu, S.Melvin, M.Shebanow, C.Chen, J.Wei)

Extending a Prolog Machine for Parallel Execution *Proceedings of the Hawaii International Conference on Systems Sciences* Honolulu, HI, January 1986 (with T. Dobry, J. Chang, A. Despain)

A C Compiler for the HPS I, a Highly Parallel Execution Engine *Proceedings of the Hawaii International Conference on Systems Sciences* Honolulu, HI, January 1986 (with M. Shebanow)

Microcode and the Protection of Intellectual Effort *Proceedings of the 18th Microprogramming Workshop* Asilomar, CA, December 1985 (with J. Ahlstrom)

Compiling Prolog into Microcode: A Case Study Using the NCR/32-000 *Proceedings of the 18th Microprogramming Workshop* Asilomar, CA, December 1985 (with B. Fagin, A. Despain, and V. Srini)

HPS, A New Microarchitecture: Rationale and Introduction *Proceedings of the 18th Microprogramming Workshop* Asilomar, CA, December 1985 (with W. Hwu and M. Shebanow)

Critical Issues Regarding HPS, A High Performance Microarchitecture *Proceedings of the 18th Microprogramming Workshop* Asilomar, CA, December 1985 (with S. Melvin, W. Hwu, and M. Shebanow)

Performance Studies of a Prolog Machine Architecture *Proceedings of the 12th Annual International Symp. on Computer Architecture* June, 1985 (with T.P.Dobry and A.M.Despain)

Aquarius - A High Performance Computing System for Symbolic/Numeric Applications, *Proceedings of Compcon85* February, 1985 (with A. M. Despain)

Design Decisions Influencing the Microarchitecture for a Prolog Machine *Proceedings of the 17th Annual Microprogramming Workshop* October, 1984 (with T. P. Dobry and A. M. Despain)

Alternative Proposals for Implementing Prolog Concurrently and Implications regarding their Respective Microarchitectures *Proceedings of the 17th Annual Microprogramming Workshop* October, 1984 (with C. Ponder)

A Comparison of Several Evolving (University) Supercomputer Architectures *Proceedings of the 4th Jerusalem Conference on Information Technology* May, 1984 (with R. Sheldon, M. Shebanow, C. Ponder, and W. Hwu)

The Aquarius Project *Proceedings of Compcon84* February, 1984 (with A. M. Despain)

Tradeoffs in the design of a system for high-level language interpretation, IEEE Intl. Conf. on Computer Design: VSLI Computers, Port Chester, New York, October 31-November 3, 1983 (with F. Colon-Osorio).

Modifications to the VAX-11/780 microarchitecture to support IEEE floating point arithmetic, 16th Annual Microprograming Workshop, IEEE Computer Society, Dowingtown, Pennsylvania, October 11-14, 1983 (with D. Aspinwall).

Improving the performance of UCSD Pascal by Microprogramming the PDP 11/60, 16th Annual Microprogramming Workshop, IEEE Computer Society, Dowingtown, Pennsylvania, October 11-14, 1983 (with M.T. Shaeffer).

Some results on the asymptotic behavior of functions on subsets of the natural numbers, (with D.F. MacAllister), 1979 Southeastern ACM Regional Meeting, April, 1979.

The linearity property and functional completeness in m-valued logic, *International Conference on Multiple Valued Logic,* Indiana University, May, 1975.

Microprogramming -- What it is, what it isn't, where it's been, where it's going, *1975 Southeastern ACM Regional Meeting,* Raleigh, N.C., April, 1975.

Non-linearity, a new necessary condition for logical completeness in k-valued logic, *Proceedings of the Sixth Annual Hawaii International Conference on Systems Sciences,* January, 1973.

Toward a characterization of logically complete switching functions in k-valued logic, *Conference Record of the 1972 Symposium on the Theory and Applications of Multiple-valued Logic Design,* Buffalo, New York, May 1972.

Injections of neighborhood size three and four on the set of configurations from the infinite one-dimensional tesselation automaton of two-state cells, *Proceedings of the Fourth Southeastern Sumposium on Systems Theory,* April, 1972.

A note on functional completeness in many-valued logic, *Sixth Annual Princeton Conference on Information Sciences and Systems,* March, 1972 (with H. El Lozy).

Synthesis of multivalued switching functions -- An approach to the completeness property in k-valued logic, *Proceedings of the Fifth Hawaii International Conference on System Sciences,* January, 1972.

The ULM-n vs. the WOS-n; Two approaches to the design of integrated circuit switching functions, *Proceedings of the Third Southeastern Symposium on Systems Theory,* April, 1971.

The effects of considering equivalence classes in computing lower bounds on the number of external interconnections for a universal logic module of *n* variables, *Fifth Annual Princeton Conference on Information Sciences and Systems,* March 25-26, 1971.

Universal logic modules with still fewer external interconnections, *Proceedings of the Fourth Hawaii International Conference on Systems Sciences,* January, 1971.

Structural properties of unbounded optimal doubly-chained trees, *Third Hawaii International Conference on Systems Sciences,* January, 1970.

A lemma relating the binary coefficients of two canonical expressions for combinatorial switching functions, *Third Hawaii International Conference on System Sciences,* January, 1970.

Optimal layout of a warehouse system with high turn-around time, *1969 National Meeting of ORSA,* November, 1969.

A complex logic module for the synthesis of combinatorial switching circuits, *Proceedings of the 1967 Spring Joint Computer Conference,* Atlantic City, April, 1967.